AT9-99-129   **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Balaram Sinharoy et al.

Serial No.:   09/548,469   Art Unit: 2183

Filed: April 13, 2000   Examiner: Aimee J. Li

For: USE OF SOFTWARE HINT FOR BRANCH PREDICTION IN THE ABSENCE OF HINT BIT IN THE BRANCH INSTRUCTION

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION - 37 CFR 1.192)

1. **Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on June 9, 2004.**

> NOTE: *"The appellant shall, within 2 months from the date of the notice of appeal under § 1.191 in an application, reissue application, or patent under reexamination, or within the time allowed for response to the action appealed from, if such time is later, file a brief in triplicate." 37 CFR 1.192(a) (emphasis added).*

2. **STATUS OF APPLICANT**

   This application is on behalf of

   ☒ other than a small entity

   ☐ small entity

   verified statement:

       ☐ attached

       ☐ already filed

3. **FEE FOR FILING APPEAL BRIEF**

   Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is:

   ☐ small entity $165.00

   ☒ other than a small entity $330.00

           Appeal Brief fee due   **$330.00**

(Page 1 of 3)

4. EXTENSION OF TERM

NOTE: The time periods set forth in 37 CFR 1.192(a) are subject to the provision of § 1.136 for patent applications. 37 CFR 1.191(d). Also see Notice of November 5, 1985 (1060 O.G. 27).

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply.

(complete (a) or (b) as applicable)

(a) ☐ Applicants petition for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

Extension (months)	Fee for other than small entity	Fee for small entity
☐ one month	$ 110.00	$ 55.00
☐ two months	$ 410.00	$ 205.00
☐ three months	$ 930.00	$ 465.00
☐ four months	$ 1,450.00	$ 725.00
Fee		

If an additional extension of time is required, please consider this a petition therefor.

(check and complete the next item, if applicable)

☐ An extension for _____ months has already been secured and the fee paid therefor of $_____ is deducted from the total fee due for the total months of extension now requested.

Extension fee due with this request $_____

or

(b) ☒ Applicants believe that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicants have inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE
The total fee due is:
Appeal Brief fee $330.00
Extension fee (if any) $0

TOTAL FEE DUE $330.00

6. FEE PAYMENT
☐ Attached is a check in the sum of $_____
☒ Charge Account No. 09-0447 (AT9-99-129) the sum of $330.00.

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

NOTE: *If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.*

☒   If any additional extension and/or fee is required, this is a request therefor and to charge Account No. <u>09-0447 (AT9-99-129)</u>.
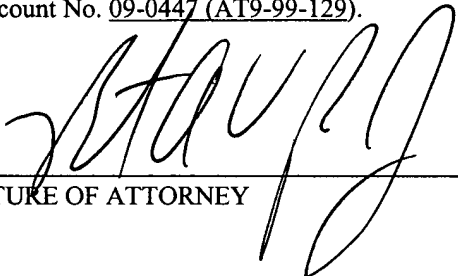
<div align="center">AND/OR</div>

☒   If any additional fee for claims is required, charge Account No. <u>09-0447 (AT9-99-129)</u>.

Reg. No.:   <u>47,159</u>

<div align="right">_____
SIGNATURE OF ATTORNEY</div>

Tel. No.:   <u>(512) 370-2832</u>

<u>Robert A. Voigt, Jr.</u>
WINSTEAD SECHREST & MINICK P.C.
P.O. Box 50784
Dallas, TX 75201

Austin_1 257267v.1

-1-

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	:	Before the Examiner:
Balaram Sinharoy et al.	:	Li, Aimee J.
Serial No.: 09/548,469	:	Group Art Unit: 2183
Filed: April 13, 2000	:	
	:	IBM Corporation
Title: USE OF SOFTWARE HINT FOR	:	Intellectual Property Law
BRANCH PREDICTION IN THE	:	11400 Burnet Road
ABSENCE OF HINT BIT IN THE	:	Austin, Texas 78758
BRANCH INSTRUCTION	:	

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I.   REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, which is the assignee of the entire right, title and interest in the above-identified patent application.

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on August 9, 2004.

Signature

Serena Beller
(Printed name of person certifying)

08/11/2004 KBETEMA1 00000055 090447   09548469
01 FC:1402   330.00 DA

II.   RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III.   STATUS OF CLAIMS

Claims 1-14, 39 and 40 are pending in the Application. Claims 1-14, 39 and 40 stand rejected.

IV.   STATUS OF AMENDMENTS

Appellants' response to the Office Action having the mailing date of September 25, 2003, has been considered, but the Examiner indicated that it did not place the application in condition for allowance because Appellants' arguments were deemed unpersuasive.

V.   SUMMARY OF INVENTION

For many applications, the compiler can often determine how a conditional branch should be predicted by the hardware at run-time. Specification, page 6, lines 7-8. For some applications, the software branch prediction can be highly accurate. Specification, page 6, lines 8-9. The software branch prediction can be very useful for microprocessors that do not have a hardware branch prediction mechanism. Specification, page 6, lines 9-10. It is also useful for improving the hardware branch prediction accuracy for some applications, by combining the software branch prediction with the hardware branch prediction mechanism through mechanisms such as an agree/disagree prediction algorithm as explained further below. Specification, page 6, lines 10-14.

Ordinarily, the branch history table (BHT) stores information about the branch's outcome. Specification, page 6, lines 15-16. For example, in a 2-bit per entry BHT implementation, each entry indicates whether the associated BHT entry should be predicted taken (1x) or not-taken (0x). Specification, page 6, lines 16-18.

When a branch is executed, if it is found to be taken, the entry is incremented (if it is already "11", then there is no change). Specification, page 6, lines 18-19. If it is found to be not-taken, the entry is decremented (if it is already "00", then there is no change). Specification, page 6, lines 19-20.

For agree/disagree prediction, instead of storing the taken/not-taken information in the BHT, the information stored is whether the branch outcome at execution was in agreement with the software branch prediction or not. Specification, page 7, lines 1-3. If the software predicted taken and the branch is actually found to be taken when it is executed, then the branch "agrees" with the software prediction. Specification, page 7, lines 3-5. Similarly, if the software prediction is not-taken and the branch is actually found to be not-taken during execution, then also the branch is considered to have "agreed" with the software prediction. Specification, page 7, lines 5-7. Otherwise, the branch "disagrees" with the software prediction. Specification, page 7, lines 7-8. When a branch is executed, its associated entry in the BHT is updated based on whether the branch "agrees" or "disagrees" with the software prediction. Specification, page 7, lines 8-9. If the branch agrees, then the entry is incremented (no change, if it is already "11"). Specification, page 7, lines 9-10. If the branch disagrees, then the entry is decremented (no change, if it is already "00"). Specification, page 7, lines 10-11. When a branch is fetched, if its associated entry in the BHT is "1x", then the branch is predicted to agree with the software prediction. Specification, page 7, lines 11-13. On the other hand, if its associated entry in the BHT is "0x", then the prediction made is opposite of what the software predicted. Specification, page 7, lines 13-15.

The primary advantage of agree/disagree prediction is that for many applications it decreases the harmful effects of aliasing in the BHT. Specification, page 7, lines 16-17.

In many architectures, the branch instructions do not have any unused or reserved bits used to provide branch prediction hints by the software. Specification, page 8, lines 1-2. Such hints can communicate to the hardware how the software

thinks the branch should be predicted. Specification, page 8, lines 2-3. Therefore, there is a need in the art for providing software branch prediction hints to the hardware. Specification, page 8, lines 3-5.

The problems outlined above may at least in part be solved in one embodiment by a method for predicting a result of a conditional branch instruction. Specification, page 25, lines 1-2. The method may include determining if a specified condition register field is used to store a branch condition of the conditional branch instruction. Specification, page 25, lines 3-4. The method may further include providing a software branch prediction of the conditional branch instruction as a function of the determination if the specified condition register field is used to store the branch condition of the conditional branch instruction. Specification, page 25, lines 5-7.

VI.   ISSUE

Are claims 1-14 and 39-40 properly rejected under 35 U.S.C. §102(e) as being as being anticipated by Henry et al. (U.S. Patent No. 6,550,004) (hereinafter "Henry")?

VII.   GROUPING OF CLAIMS
Claims 1 and 8 form a first group.
Claims 2 and 9 form a second group.
Claims 3 and 10 form a third group.
Claims 4 and 11 form a fourth group.
Claims 5 and 12 form a fifth group.
Claims 6, 7, 13 and 14 form a sixth group.
Claims 39 and 40 should not be grouped together and should be considered separately.
The reasons for these groupings are set forth in Appellants' arguments in Section VIII.

VIII. ARGUMENT

The Examiner has rejected claims 1-14 and 39-40 under 35 U.S.C. §102(e) as being anticipated by Henry. Paper No. 10, page 2. Appellants respectfully traverse and assert that claims 1-14 and 39-40 are not properly rejected under 35 U.S.C. §102(e) as being anticipated by Henry for at least the reasons stated below.

For a claim to be anticipated under 35 U.S.C. §102, each and every claim limitation <u>must</u> be found within the cited prior art reference and arranged as required by the claim. M.P.E.P. §2131.

Appellants respectfully assert that Henry does not disclose "providing a software branch prediction of the conditional branch instruction as a function of the determination if the specified condition register field is used to store the branch condition of the conditional branch condition" as recited in claim 1 and similarly in claim 8. The Examiner cites lines 13-14 of the abstract; column 4, lines 49-52; column 5, lines 8-12 and 35-38; column 9, lines 31-44 and Figure 2 of Henry as disclosing the above-cited claim limitation. Paper No. 10, pages 2-3. Appellants respectfully traverse and assert that Henry instead discloses a static predictor, static predictor 222, that receives three inputs and predicts the outcome of conditional branch instructions based upon the three inputs. One of the three inputs comprises a conditional branch instruction test type. The test type specifies a condition upon which the branch instruction will be taken or not taken. This language is not the same as providing a branch prediction as a function of the determination if a specified condition register field is used to store the branch condition of the conditional branch condition. There is no language in Henry that discloses providing a prediction if a condition register field is used to store a branch condition. Further, there is no language in Henry that discloses a software branch prediction. Thus, Henry does not disclose all of the limitations of claims 1 and 8, and thus Henry does not anticipate claims 1 and 8. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner states that Henry (column 9, lines 36-37) discloses x86 conditional jump instruction test types. Paper No. 10, page 6. The Examiner further states that Henry (column 9, lines 37-40) discloses that the x86 conditional jump instruction test types include conditions based upon the carry, overflow, zero, parity and sign flags of the x86 FLAGS register. Paper No. 10, page 6. However, the Examiner has not provided any objective evidence to support the assertion that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses providing a prediction if a condition register field is used to store a branch condition. There is no language in Henry that suggests that a prediction is provided based on whether a field is used to store a branch condition. Further, the disclosure of an x86 FLAGS register is not related to a software branch prediction. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses providing a prediction if a condition register field is used to store a branch condition. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses providing a prediction if a condition register field is used to store a branch condition, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claims 1 and 8. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "determining if the conditional branch instruction is positioned at a specified address in a sequence of instructions being executed" as recited in claim 39. The Examiner cites column 7,

lines 5-6 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 5. Appellants respectfully traverse and assert that Henry instead discloses a branch predictor, branch predictor 103, that receives the address of branch instructions from the instruction pointer. This language has no relevancy to determining if an instruction is positioned at an address in a sequence of instructions being executed. Thus, Henry does not disclose all of the limitations of claim 39, and thus Henry does not anticipate claim 39. M.P.E.P. §2131.

The Examiner further states in connection with the rejection of the above-cited claim limitation:

> Henry has taught in column 7, lines 5-9; column 8, lines 14-16 and 31-41; and shown in Figures 1 and 2 that a branch prediction is based upon the branch instruction address. The branch instruction address identifies the location a branch instruction in a sequence of instructions. As is known in the art, the instruction pointer register (IP register) contains the current position of the instruction being executed in a sequence of instructions. This IP Register sends the position of the instruction via a signal to the branch predictor so that the branch predictor will make a prediction based upon the branch instruction address. Paper No. 10, pages 7-8.

Appellants respectfully assert that the Examiner has not provided any objective evidence for asserting that a pointer that points to the current instruction being executed discloses determining if a conditional branch instruction is positioned at a specified address. Neither has the Examiner provided any objective evidence for asserting that a pointer that points to the current instruction being executed discloses determining if a conditional branch instruction is positioned at a specified address in a sequence of instructions being executed. There is no language in Henry that suggests determining if a conditional branch instruction is positioned at a specified address. Neither is there any language in Henry that suggests determining if a conditional branch instruction is positioned at a specified address in a sequence of instructions being executed. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that a pointer that points to the current instruction being executed discloses determining if a conditional branch instruction is positioned

at a specified address in a sequence of instructions being executed. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that a pointer that points to the current instruction being executed discloses determining if a conditional branch instruction is positioned at a specified address in a sequence of instructions being executed, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claim 39. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "predicting whether the conditional branch instruction will be taken or not taken as a function of the position of the specified address" as recited in claim 39. The Examiner cites column 7, lines 5-8 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 5. Appellants respectfully traverse and assert that Henry instead discloses a branch predictor, branch predictor 103, that receives the address of branch instructions from an instruction pointer via a signal and makes a prediction of the outcome of the branch instruction based upon the branch instruction address. This language is not the same as predicting whether a conditional branch instruction will be taken or not taken as a function of a <u>position</u> of an address. Henry simply discloses predicting the outcome of the branch instruction based upon the branch instruction address but not predicting the outcome of the branch instruction based upon a <u>position of the address</u>. Thus, Henry does not disclose all of the limitations of claim 39, and thus Henry does not anticipate claim 39. M.P.E.P. §2131.

The Examiner further states in connection with the rejection of the above-cited claim limitation the same passage quoted by Appellants on page 7 of this Appeal Brief. Appellants will not restate it for the sake of brevity. Appellants respectfully assert that the Examiner has not provided any objective evidence for asserting that a

pointer that points to the current instruction being executed discloses predicting whether a conditional branch instruction will be taken or not taken as a function of the position of the specified address. There is no language in Henry that suggests predicting whether a conditional branch instruction will be taken or not taken as a function of a position of a specified address. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that a pointer that points to the current instruction being executed discloses predicting whether a conditional branch instruction will be taken or not taken as a function of the position of the specified address. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that a pointer that points to the current instruction being executed discloses predicting whether a conditional branch instruction will be taken or not taken as a function of the position of the specified address, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claim 39. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "wherein the software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is used to store the branch condition of the conditional branch instruction" as recited in claim 2 and similarly in claim 9. The Examiner cites lines 13-14 of the abstract; column 4, lines 49-52; column 5, lines 8-12 and 35-38; column 9, lines 31-44 and Figure 2 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 3. Appellants respectfully traverse and assert that Henry instead discloses a static predictor, static predictor 222, that receives three inputs and predicts the outcome of conditional branch instructions based upon the three inputs. One of the three inputs comprises a conditional branch instruction test type. The test type specifies a condition upon which the branch instruction will be taken or not taken. This language is not the same as predicting a conditional

branch instruction will be taken if a specified condition register field is used to store a branch condition. There is no language in Henry in basing a prediction on whether a particular condition register field is used to store a branch condition. Further, as stated above, there is no language in Henry that discloses a software branch prediction. Thus, Henry does not disclose all of the limitations of claims 2 and 9, and thus Henry does not anticipate claims 2 and 9. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner states that Henry (column 9, lines 36-37) discloses x86 conditional jump instruction test types. Paper No. 10, page 6. The Examiner further states that Henry (column 9, lines 37-40) discloses that the x86 conditional jump instruction test types include conditions based upon the carry, overflow, zero, parity and sign flags of the x86 FLAGS register. Paper No. 10, page 6. However, the Examiner has not provided any objective evidence to support the assertion that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses predicting a conditional branch instruction will be taken if a specified condition register field is used to store a branch condition. There is no language in Henry that suggests predicting a conditional branch instruction will be taken if a specified condition register field is used to store the branch condition of the conditional branch instruction. Further, the disclosure of an x86 FLAGS register is not related to a software branch prediction. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses predicting a conditional branch instruction will be taken if a specified condition register field is used to store a branch condition. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses predicting a conditional branch instruction will be taken if a specified condition register field is used to store a branch

condition, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claims 2 and 9. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "wherein the software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction" as recited in claim 3 and similarly in claim 10. The Examiner cites lines 13-14 of the abstract; column 4, lines 49-52; column 5, lines 8-12 and 35-38; column 9, lines 31-44 and Figure 2 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 3. Appellants respectfully traverse and assert that Henry instead discloses a static predictor, static predictor 222, that receives three inputs and predicts the outcome of conditional branch instructions based upon the three inputs. One of the three inputs comprises a conditional branch instruction test type. The test type specifies a condition upon which the branch instruction will be taken or not taken. This language is not the same as predicting a conditional branch instruction will not be taken if a specified condition register field is not used to store a branch condition. There is no language in Henry in basing a prediction on whether a particular condition register field is not used to store a branch condition. Further, as stated above, there is no language in Henry that discloses a software branch prediction. Thus, Henry does not disclose all of the limitations of claims 3 and 10, and thus Henry does not anticipate claims 3 and 10. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner states:

> In this instance, Henry's predictor bases a prediction off of the instruction test type, which includes the flag fields found within the x86 FLAGS register. As stated by Henry in column 7, lines 18-20, the static prediction is based upon the test condition test type, which

indicates whether a specified condition register field is used or not used by the branch conditional. For example, the x86 conditional jump instructions, which are the same as conditional branch instructions, referred to by Henry in column 9, lines 36-37 include instructions with test types which refer to specific fields within the condition register, such as a jump on carry which refers to the carry field within the condition register. The predictor of Henry would determine a branch prediction based upon whether or not the branch test type is a jump on carry or not. Also, the x86 conditional jump instructions include instructions with test types that do not refer to specific fields within the condition register, such as a jump if greater. These type of conditional branch test types do not refer to any of the condition register fields in the x86 FLAGS register, and Henry's predictor will predict them as taken or not taken accordingly. Please see Intel's Pentium® Processor Family Developer's Manual Volume 3: Architecture and Programming Manual, © 1995 pages 4-25 to 4-26 and D-1 to D-2 for more information. Whether Henry predicts a branch test type as taken or not taken in specific instances does not matter, since this does not change the functionality of the device. Paper No. 10, pages 8-9.

Appellants respectfully disagree with the Examiner's interpretation of Henry. Henry does disclose that static predictor 222 generates its prediction based in part on the test type. However, there is no basis in fact and/or technical reasoning to interpret generating a prediction based in part on a test type as disclosing whether a specified condition register field is used or not used by the branch conditional as asserted by the Examiner. Instead, Henry is referring to the flag states in the x86 FLAGS register that includes information that may be used by static predictor 222 to generate its prediction. Column 9, lines 38-40. For example, the carry flag (one of the flags in the x86 FLAGS register as indicated by Intel's Pentium® Processor Family Developer's Manual Volume 3: Architecture and Programming Manual, © 1995 pages 4-25 to 4-26 cited by the Examiner) indicates if an overflow has occurred when using unsigned numbers.[1] The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion generating a prediction based in part on a test type discloses whether a specified condition register field is used or not used by

---

[1] For example, suppose in an 8 bit instruction the value of 1 is added to 255. Since 255 is the data limit for a byte, the result cannot be 256. Consequently, the result will be 0 along with setting the carry flag.

the branch conditional. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner <u>must</u> provide extrinsic evidence that must make clear that generating a prediction based in part on a test type discloses whether a specified condition register field is used or not used by the branch conditional, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claims 3 and 10. M.P.E.P. §2131.

Furthermore, Appellants would like to clarify the Examiner's statement that the static predictor in Henry makes its prediction based on whether or not the branch test type is a jump carry or not. As understood by Appellants, the static predictor in Henry generates its prediction based in part on the states of the flag, e.g., carry flag, in the FLAGS register as discussed on pages 4-25 to 4-26 in the Intel's Pentium® Processor Family Developer's Manual Volume 3. This language is not the same as predicting a branch instruction will not be taken if a field is not used to store a branch condition of the conditional branch instruction. Thus, Henry does not disclose all of the limitations of claims 3 and 10, and thus Henry does not anticipate claims 3 and 10. M.P.E.P. §2131.

Furthermore, with respect to the Examiner's statement that "whether Henry predicts a branch test type as taken or not taken in specific instances does not matter, since this does not change the functionality of the device", Appellants respectfully traverse the implied assertion that predicting whether a conditional branch instruction is taken or not taken if a field is not used to store a branch condition of the conditional branch instruction does not matter since this does not change the functionality of the device. The Examiner must not ignore language in the claims. All words in a claim must be considered in judging the patentability of that claim against the prior art. *In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970). The

Examiner must identify a reference that discloses where a software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction. M.P.E.P. §2131. Since the Examiner has not shown that Henry discloses all of the limitations of claims 3 and 10, the Examiner has not presented a *prima facie* case of anticipation for rejecting claims 3 and 10. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "wherein the software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is used to store the branch condition of the conditional branch instruction" as recited in claim 4 and similarly in claim 11. The Examiner cites lines 13-14 of the abstract; column 4, lines 49-52; column 5, lines 8-12 and 35-38; column 9, lines 31-44 and Figure 2 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 4. Appellants respectfully traverse and assert that Henry instead discloses a static predictor, static predictor 222, that receives three inputs and predicts the outcome of conditional branch instructions based upon the three inputs. One of the three inputs comprises a conditional branch instruction test type. The test type specifies a condition upon which the branch instruction will be taken or not taken. This language is not the same as predicting a conditional branch instruction will not be taken if a specified condition register field is used to store a branch condition. There is no language in Henry in basing a prediction on whether a particular condition register field is not used to store a branch condition. Further, as stated above, there is no language in Henry that discloses a software branch prediction. Thus, Henry does not disclose all of the limitations of claims 4 and 11, and thus Henry does not anticipate claims 4 and 11. M.P.E.P. §2131.

Further, in connection with the rejection of the above-cited claim limitation, the Examiner states that Henry (column 9, lines 36-37) discloses x86 conditional jump instruction test types. Paper No. 10, page 6. The Examiner further states that Henry (column 9, lines 37-40) discloses that the x86 conditional jump instruction test types include conditions based upon the carry, overflow, zero, parity and sign flags of

the x86 FLAGS register. Paper No. 10, page 6. However, the Examiner has not provided any objective evidence to support the assertion that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses predicting a conditional branch instruction will be not taken if a specified condition register field is used to store a branch condition. There is no language in Henry that suggests predicting a conditional branch instruction will be not taken if a specified condition register field is used to store the branch condition of the conditional branch instruction. Further, the disclosure of an x86 FLAGS register is not related to a software branch prediction. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses predicting a conditional branch instruction will be not taken if a specified condition register field is used to store a branch condition. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner <u>must</u> provide extrinsic evidence that must make clear that a register containing fields where each field may represent a particular flag and where the bit(s) in each field may represent a state of a particular flag discloses predicting a conditional branch instruction will be not taken if a specified condition register field is used to store a branch condition, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claims 4 and 11. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "wherein the software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction" as recited in claim 5 and similarly in claim 12. The Examiner cites lines 13-14 of the abstract; column 4, lines 49-52; column 5, lines 8-

12 and 35-38; column 9, lines 31-44 and Figure 2 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 4. Appellants respectfully traverse and assert that Henry instead discloses a static predictor, static predictor 222, that receives three inputs and predicts the outcome of conditional branch instructions based upon the three inputs. One of the three inputs comprises a conditional branch instruction test type. The test type specifies a condition upon which the branch instruction will be taken or not taken. This language is not the same as predicting a conditional branch instruction will be taken if a specified condition register field is not used to store a branch condition. There is no language in Henry in basing a prediction on whether a particular condition register field is not used to store a branch condition. Further, as stated above, there is no language in Henry that discloses a software branch prediction. Thus, Henry does not disclose all of the limitations of claims 5 and 12, and thus Henry does not anticipate claims 5 and 12. M.P.E.P. §2131.

The Examiner further states in connection with the rejection of the above-cited claim limitation the same passage quoted by Appellants on pages 11-12 of this Appeal Brief. Appellants will not restate it for the sake of brevity. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that generating a prediction based in part on a test type discloses whether a specified condition register field is used or not used by the branch conditional. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner <u>must</u> provide extrinsic evidence that must make clear that generating a prediction based in part on a test type discloses whether a specified condition register field is used or not used by the branch conditional, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claims 5 and 12. M.P.E.P. §2131.

Furthermore, Appellants would like to clarify the Examiner's statement that the static predictor in Henry makes its prediction based on whether or not the branch test type is a jump carry or not. As understood by Appellants, the static predictor in Henry generates its prediction based in part on the states of the flag, e.g., carry flag, in the FLAGS register as discussed on pages 4-25 to 4-26 in the Intel's Pentium® Processor Family Developer's Manual Volume 3. This language is not the same as predicting a branch instruction will be taken if a field is not used to store a branch condition of the conditional branch instruction. Thus, Henry does not disclose all of the limitations of claims 5 and 12, and thus Henry does not anticipate claims 5 and 12. M.P.E.P. §2131.

Furthermore, with respect to the Examiner's statement that "whether Henry predicts a branch test type as taken or not taken in specific instances does not matter, since this does not change the functionality of the device", Appellants respectfully traverse the implied assertion that predicting whether a conditional branch instruction is taken or not taken if a field is not used to store a branch condition of the conditional branch instruction does not matter since this does not change the functionality of the device. The Examiner must not ignore language in the claims. All words in a claim must be considered in judging the patentability of that claim against the prior art. *In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970). The Examiner must identify a reference that discloses where a software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction. M.P.E.P. §2131. Since the Examiner has not shown that Henry discloses all of the limitations of claims 5 and 12, Henry does not anticipate claims 5 and 12. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "wherein the specified condition register field is N, where N is an integer" as recited in claim 6 and similarly in claim 13. Appellants further assert that Henry does not disclose "wherein the specified condition register field is a multiple of N" as recited in claim 7 and similarly

in claim 14. The Examiner cites column 7, lines 36-38 and column 9, lines 31-44 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 4. Appellants respectfully traverse and assert that Henry instead discloses a register file, register file 105, that includes a status flags register that is used in determining whether branch conditions have been satisfied. While the status flags register is used in determining whether branch conditions have been satisfied, presumably based on the state of flags stored in such a register, Henry does not disclose that a prediction is a function of whether a specified field in the status flags register was used to store a branch condition. Further, Henry does not disclose that such a specified field is N or a multiple of N. Thus, Henry does not disclose all of the limitations of claims 6, 7, 13 and 14, and thus Henry does not anticipate claims 6, 7, 13 and 14. M.P.E.P. §2131.

In response to Appellants' argument that Henry does not anticipate claims 6, 7 13 and 14, the Examiner states that each field within a condition register is referenced by a computer by a bit location and consequently Henry discloses the above-cited claim limitations. Paper No. 10, page 9. However, the Examiner is ignoring language in the claims. All words in a claim must be considered in judging the patentability of that claim against the prior art. *In re Wilson*, 424 F.2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970). The Examiner must identify a reference that discloses that a prediction is a function of whether a <u>specified condition register field is used to store a branch condition of the conditional branch instruction</u> where the specified condition register field is N, where N is an integer, or where the specified condition register field is a multiple of N. Since the Examiner has not shown that Henry discloses all of the limitations of claims 6, 7 13 and 14, Henry does not anticipate claims 6, 7 13 and 14. M.P.E.P. §2131.

Appellants further assert that Henry does not disclose "wherein the predicting program step will predict taken if the specified address is a multiple of specified number N" as recited in claim 40. The Examiner cites column 8, lines 14-16 and 31-41 and column 7, lines 5-8 of Henry as disclosing the above-cited claim limitation. Paper No. 10, page 5. Appellants respectfully traverse and assert that Henry instead

18

discloses a branch predictor, branch predictor 103, that receives the address of branch instructions from an instruction pointer via a signal and makes a prediction of the outcome of the branch instruction based upon the branch instruction address. Henry further discloses dynamic predictors, predictors 202, 204, that receive an address of conditional branch instructions from the instruction pointer register. Henry further discloses that dynamic predictor 202 comprises history table X 302 and that dynamic predictor 204 comprises history table Y 304. This language is not the same as predicting whether a conditional branch instruction will be taken or not taken as a function of a position of an address. Henry simply discloses predicting the outcome of the branch instruction based upon the branch instruction address but not predicting the outcome of the branch instruction based upon a position of the address. Further, Henry does not disclose predicting a conditional branch instruction will be taken if the address is a multiple of specified number N. Thus, Henry does not disclose all of the limitations of claim 40, and thus Henry does not anticipate claim 40. M.P.E.P. §2131.

In response to Appellants' argument that Henry does not anticipate claim 40, the Examiner states:

> As shown above, Henry has taught a prediction is based upon the branch address. The global history table is indexed, which provides a dynamic branch prediction in the system, by hashing the branch address with the global history register to index the actual history tables. Hashing the branch address with the global history register means, in essence, numerically manipulating the address, which includes determining whether the branch address is a multiple of the global history register. Paper No. 10, page 10.

Appellants respectfully traverse the assertion that hashing a branch address with a global history register corresponds to manipulating an address which includes determining whether the branch address is a multiple of the global history register. The Examiner has not provided any objective evidence to support her assertion but instead relies upon her own subjective opinion. Instead, Henry discloses that dynamic predictor X 202 is configured to hash the entire global history in generating

an index into the history table. Column 9, lines 59-61. Henry further discloses that dynamic predictor Y 204 is configured to hash only two bits of the global history in generating an index into the history table. Column 10, lines 1-3. Hence, Henry discloses dynamic predictors hashing two bits or the entire global history to generate a value that is used to index into a history table. This value that results from hashing is not related to a specified address in a sequence of instructions being executed. The Examiner must provide a basis in fact and/or technical reasoning to reasonably support the assertion that hashing two bits or the entire global history to generate a value that is used to index into a history table discloses determining if a conditional branch instruction is positioned at a specified address in a sequence of instructions being executed where the conditional branch instruction will be predicted taken if the specified address is a multiple of specified number N. *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). That is, the Examiner must provide extrinsic evidence that must make clear that hashing two bits or the entire global history to generate a value that is used to index into a history table discloses determining if a conditional branch instruction is positioned at a specified address in a sequence of instructions being executed where the conditional branch instruction will be predicted taken if the specified address is a multiple of specified number N, and that it be so recognized for persons of ordinary skill. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999). Therefore, the Examiner must support her assertion with objective evidence meeting the above requirements. The Examiner has not provided any objective evidence to support her assertion, and therefore the Examiner has not presented a *prima facie* case of anticipation for rejecting claim 40. M.P.E.P. §2131.

As a result of the foregoing, Appellants respectfully assert that not each and every claim limitation was found within the cited prior art reference and thus claims 1-14 and 39-40 are not anticipated by Henry.

IX.   <u>CONCLUSION</u>

For the reasons noted above, the rejections of claims 1-14 and 39-40 are in error. Appellants respectfully request reversal of the rejections and allowance of claims 1-14 and 39-40.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Appellants

By:_____

Kelly K. Kordzik
Reg. No. 36,571
Robert A. Voigt, Jr.
Reg. No. 47,159

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

APPENDIX

1.   A method for predicting a result of a conditional branch instruction, comprising the steps of:

determining if a specified condition register field is used to store a branch condition of the conditional branch instruction; and

providing a software branch prediction of the conditional branch instruction as a function of the determination if the specified condition register field is used to store the branch condition of the conditional branch instruction.

2.   The method as recited in claim 1, wherein the software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is used to store the branch condition of the conditional branch instruction.

3.   The method as recited in claim 2, wherein the software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction.

4.   The method as recited in claim 1, wherein the software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is used to store the branch condition of the conditional branch instruction.

5.   The method as recited in claim 4, wherein the software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction.

6.   The method as recited in claim 1, wherein the specified condition register field is N, where N is an integer.

7.   The method as recited in claim 6, wherein the specified condition register field is a multiple of N.

8.   A processor comprising:

an instruction fetch unit for fetching a conditional branch instruction;

circuitry for determining if a specified condition register field is used to store a branch condition of the conditional branch instruction; and

circuitry for providing a software branch prediction of the conditional branch instruction as a function of the determination if the specified condition register field is used to store the branch condition of the conditional branch instruction.

9.   The processor as recited in claim 8, wherein the software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is used to store the branch condition of the conditional branch instruction.

10.   The processor as recited in claim 9, wherein the software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction.

11.   The processor as recited in claim 8, wherein the software branch prediction predicts that the conditional branch instruction will be not taken if the specified condition register field is used to store the branch condition of the conditional branch instruction.

12.   The processor as recited in claim 11, wherein the software branch prediction predicts that the conditional branch instruction will be taken if the specified condition register field is not used to store the branch condition of the conditional branch instruction.

13.   The processor as recited in claim 8, wherein the specified condition register field is N, where N is an integer.

14.   The processor as recited in claim 13, wherein the specified condition register field is a multiple of N.

39.   A data processing system for predicting whether a conditional branch instruction will be taken or not taken, the data processing system comprising the program steps of:

determining if the conditional branch instruction is positioned at a specified address in a sequence of instructions being executed; and

predicting whether the conditional branch instruction will be taken or not taken as a function of the position of the specified address.

40.   The data processing system as recited in claim 39, wherein the predicting program step will predict taken if the specified address is a multiple of specified number N.

Austin_1 256158v.1

24